THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES
Attorney Docket № 13940US02 (BU 2546.1)

In re Application of:

Scott S. McDaniel

Serial No.: 10/643,331

Filing Date: August 19, 2003

For: A SYSTEM AND METHOD FOR TRANSFERRING DATA OVER A REMOTE DIRECT MEMORY ACCESS (RDMA) NETWORK

Examiner: BELANI, KISHIN G

Group Art Unit No.: 2443

Confirmation No.: 3987

Customer No.: 23446

Electronically filed on 4-FEB-2011

REPLY BRIEF

MS: APPEAL BRIEF-PATENTS
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with 37 CFR 41.41, the Appellant submits this Reply Brief in response to the Examiner's Answer mailed on December 7, 2010, **with a two-month period of reply expiring on February 7, 2011**. Claims 1-25 are pending in the present Application. The Appellant has responded to the Examiner in the Examiner's Answer, as found in the following Argument section.

As may be verified in his Final Office Action dated April 19, 2010 ("Final Office Action"), the Examiner had previously rejected all pending claims 1-25.

Claims 1-5, 10 and 16 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,790,804, by Osborne.[1]

Claims 6-9, 11 and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 6,304,910, by Roach et al.[2]

Claims 12-15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 6,304,910, by Roach et al., and further in view of U.S. Patent No. 6,421,742, by Tillier.[3]

Claim 17 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 7,376,755, by Pandya.[4]

Claims 18, 20 and 22-23 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 7,376,755, by Pandya, and further in view of U.S. Patent No. 6,421,742, by Tillier.[5]

Claim 19 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 7,376,755, by

[1] *See* Final Office Action at pages 2-6.
[2] *See* Final Office Action at pages 7-14.
[3] *See* Final Office Action at pages 14-16.
[4] *See* Final Office Action at pages 16-18.
[5] *See* Final Office Action at pages 18-20.

Pandya, further in view of U.S. Patent No. 6,421,742, by Tillier, and still further in view of U.S. Patent No. 6,304,910, by Roach et al.[6]

Claim 21 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 7,376,755, by Pandya, further in view of U.S. Patent No. 6,421,742, by Tillier, and still further in view of U.S. Patent No. 5,991,797, by Futral et al.[7]

Claim 25 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,790,804, by Osborne, in view of U.S. Patent No. 7,376,755, by Pandya, and further in view of U.S. Patent No. 6,304,910, by Roach et al.[8]

To aid the Board in identifying corresponding arguments, the Appellant has used the same headings in the Argument section of this Reply Brief as the headings found in the Appellant's corresponding Brief on Appeal. The Brief on Appeal has a date of deposit of September 27, 2010.

STATUS OF THE CLAIMS

Claims 1-25 were finally rejected in the Final Office Action mailed April 19, 2010. Pending claims 1-25 are the subject of this appeal.

[6] *See* Final Office Action at pages 20-21.
[7] *See* Final Office Action at pages 21-22.
[8] *See* Final Office Action at pages 23-26.

ARGUMENT

The Appellant respectfully traverses the rejections of claims 1-25 at least based on the following arguments made in the Brief on Appeal.

I. Claims 1-5, 10 and 16 Are Not Anticipated by Osborne

Claims 1-5, 10 and 16 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Osborne. The Appellant stands by the arguments made in the corresponding sections of the Brief on Appeal as set forth in further detail below.

A. Rejection of Independent Claim 1

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer sets forth four arguments in response to the Appellant's arguments in the corresponding section of the Brief on Appeal.

Regarding the first argument in the Examiner's Answer, the Examiner's Answer states the following:

> The examiner respectfully disagrees with the appellants' argument that the cited reference of Osborne does not disclose the details of the **initiation process of a one-shot RDMA** comprising communicating a single command message that includes buffer command information, and a write command to write a send command. It is said that a picture is worth a thousand words. In the cited Osborne prior art, Fig. 1 shows all the claimed details of the **initiation process of a one-shot RDMA**, by setting up a single (i.e., one-shot) "send" command that includes a first parameter

> "ID" providing the destination address of the receiving node 52, a second parameter "Source Address" which is a pointer to a buffer or buffers that hold the message data to be transferred to the receiving node 52, and a third parameter "Size" that gives the amount of data to be transferred. **The setting up of the send command and initializing the values of the three parameters of the command <u>before the actual transfer of the message data</u> corresponds to the initiation process recited in claim 1.**
>
> As Fig. 1 clearly shows, the sending node 50 and receiving node 52 are remote to each other, connected by network 82. Further, as disclosed in the cited Osborne reference (in column 7, lines 15-45), **the operating system may set up a Direct Memory Access (DMA) with the network interface 84 to transfer the message data across network 82** to the memory buffer 74 of the receiving node 52. Thus, Osborne reference does disclose the details of the **<u>initiation process of a one-shot RDMA</u>**.
>
> Furthermore, Osborne reference (in column 7, lines 25-27 teaches copying (i.e. writing) the send message from application memory 66 to the operating system memory 62 of the sending node. The examiner has therefore shown that Osborne teaches every claim element related to the **<u>initiation process of a one shot RDMA</u>** recited in appellant's claim 1.[9]

The Appellant first notes that as shown above, the Examiner's Answer repeatedly alleges that Osborne allegedly discloses an "initiation process of a one-shot RDMA." However, Appellant's claims are directed to "**a <u>one-shot initiation process</u> of an <u>RDMA operation</u> [that] is <u>performed between the driver and the NIC of the host</u>**," not an "initiation process of a one-shot RDMA" (whatever that may be). The repeated use of "initiation process of a one-shot RDMA" in the Examiner's Answer illustrates the complete lack of understanding of the Appellant's claimed subject matter. Specifically, as discussed in the Appellant's Brief on Appeal, in prior art systems, in performing an initiation process of an RDMA operation, **<u>multiple</u> buffer command messages** (e.g.,

[9] Examiner's Answer, Page 29, Line 2 – Page 30, Line 3 (emphasis added).

command to register pinned down memory buffers into a region, binding a portion of the pinned buffer to an STag value, etc.) are sent from a driver of a host to a NIC of a host.[10] Additionally, a write command instructing the NIC to write a second command is sent from a driver of a host to a NIC of a host.[11] In contrast to the prior art, Appellant's claims are directed towards "a **one-shot** initiation process of an RDMA operation [that] is **performed between the driver and the NIC of the host, the one-shot initiation process comprising communicating a single command message comprising: buffer command information, and a write command to write a send command.**"

As discussed in the Appellant's Brief on Appeal, the Appellant notes that the cited references do not even describe the initiation process of an RDMA operation being performed between the driver and the NIC of the host. In the May 26, 2010 Appellant-initiated Examiner interview, the Examiner argued that an initiation process must occur and does not explicitly need to be disclosed because it is obvious (despite the rejection of independent claim 1 being under 35 U.S.C. 102(b)). However, as noted by the Appellant's representative in the interview, although an initiation process may occur in Osborne, the reference itself does not disclose the details of the initiation process and thus cannot disclose the Appellant's one-shot initiation process set forth in independent claim 1.

Second, as shown above, the Examiner's Answer argues that "[t]he setting up of the send command and initializing the values of the three parameters of the command

[10] *See e.g.*, Appellant's Prior Art Figure 3 (270, 290) and Appellant's Specification, Paragraph [12].
[11] *See e.g.*, Appellant's Prior Art Figure 3 (310) and Appellant's Specification, Paragraph [12].

before the actual transfer of the message data corresponds to the initiation process recited in claim 1." However, as discussed in Osborne at Column 7, Lines 14-36 and as illustrated in Osborne's Figures 1-2, at step 99, application 58 sends operating system 56 a send command 67 that identifies the message data in application memory 66 to copy to operating system message buffers 62 and to send across network 82 to receiving node 52. As such, the send command is "set up" (i.e., determination of the three parameters) at the application 58 prior to sending the send command 67 to the operating system 56. Thus, if the Examiner is interpreting that the application 58 "setting up" the send command 67 is the Appellant's one-shot initiation process of an RDMA operation (which is clearly not a one-shot initiation process), the Appellant notes that Osborne clearly fails to teach "a one-shot initiation process of an RDMA operation is **<u>performed between the driver and the NIC of the host</u>**," as set forth in Appellant's independent claim 1.

Further, Osborne's operating system 56 merely uses the send command 67 to identify the data to send across the network 82.[12] After receiving the send command 67, the operating system begins the actual transfer of message data across the network 82 to receiving node 52.[13] Nowhere in Osborne is there any disclosure regarding the send command 67 being sent between the processor 54 and the network interface 84. Also, the Examiner interprets Osborne's processor 54 to be a driver and Osborne's

[12] *See e.g.*, Osborne, Figures 1 (56), Figure 2 (100, 102, 104), Column 7, Lines 25-36.
[13] *See e.g.*, Osborne, Figures 1 (56), Figure 2 (100, 102, 104), Column 7, Lines 25-36.

network interface 84 to be a NIC.[14] As such, if the Examiner is interpreting the Appellant's "one-shot initiation process of an RDMA operation" to be Osborne's sending the send command 67 between the application 58 and the operating system 62 and the use of the send command by the operating system 62 (which is clearly not a one-shot initiation process), the Appellant notes that Osborne clearly fails to teach that "a one-shot initiation process of an RDMA operation is **performed between the driver and the NIC of the host**," as set forth in Appellant's independent claim 1.

Third, as shown above, the Examiner's Answer argues that Osborne's Column 7, Lines 15-45 teaches that "the operating system may set up a Direct Memory Access (DMA) with the network interface 84 to transfer the message data across network 82 to the memory buffer 74 of the receiving node 52." However, the Examiner's Answer mischaracterizes Osborne's disclosure. Specifically, Column 7, Lines 38-45 of Osborne refers to handling of message data received over network 82 **by the receiving node 52**. More specifically, Column 7, Lines 42-45 of Osborne refers to an embodiment of Osborne where the operating system 72 of the receiving node 52 may set up a direct memory access (DMA) with the network interface 86 of the receiving node 52 to extract and copy the message received over the network 82 at the receiving node 52 to the message buffer 74 of the receiving node 74. As such, the allegation in the Examiner's Answer that an operating system may set up a DMA with a network interface 84 at the sending node 50 to transfer message data across network 82 to the memory buffer 74 of the receiving node 52 is not taught by Osborne. Further, Osborne's teaching related

[14] *See e.g.*, Examiner's Answer, Page 4, Lines 4-7.

to setting up a DMA so that a message received over a network at a network interface 86 of a receiving node 52 may be extracted and copied to a message buffer 74 at the receiving node is wholly unrelated to a one-shot initiation process of an RDMA operation being performed between the driver and the NIC of the host.

Fourth, as shown above, the Examiner's Answer argues that Column 7, Lines 25-27 of Osborne "teaches copying (i.e. writing) the send message from application memory 66 to the operating system memory 62 of the sending node.". However, even if copying message data from application memory 66 to operating system memory 62 could be considered writing a message, such disclosure is different than "a write command to write a send command," as set forth in Appellant's independent claim 1. For example, Osborne's message(s) sent over the network 82 is the actual data transfer, whereas in reference to an RDMA operation as claimed by the Appellant, one of ordinary skill in the art would understand that the send command sent from the sending host to the receiving host advertises the buffers at the sending host so that the receiving host can post RDMA read commands for pulling data from the sending host to the receiving host.[15] In other words, Osborne's copying of message data from application memory 66 to operating system memory 62 and the sending of the copied message data across network 82 is different than the Appellant's claimed "a write command to write a send command" in connection with an RDMA operation.

[15] *See e.g.*, Appellant's Specification, Figure 3 (320, 350), Page 5, Lines 6-16 and Figure 4 (520, 550), Page 12, Lines 2-14.

Regarding the second argument in the Examiner's Answer, the Examiner's Answer states the following:

> The examiner begs to differ with the appellants' argument that nowhere in Osborne is there any specific disclosure regarding anything being sent between Osborne's processor 54 and network interface 84. By looking at the direction of arrows (representing flow of data) in Fig. 1, one would notice that data flows from application 1 at the sending node via processor 54 (driver) of the operating system 56 and NIC 84, across network 82 to the NIC 86 of the receiving node 52, and then via processor 70 (driver) of the operating system 72 to the memory of application 1 of the receiving node 52. That is what an RDMA operation involves.[16]

As acknowledged by the Examiner's Answer, the only disclosure in Osborne of anything being passed between Osborne's processor 54 (which the Examiner alleges is the Appellant's driver) and Osborne's network interface 84 (which the Examiner alleges is the Appellant's NIC) is an arrow shown in Figure 1. Nothing in Osborne teaches that the arrow passing through Osborne's processor 54 to Osborne's network interface 84 is a "one-shot initiation process comprising communicating a single command message comprising: buffer command information, and a write command to write a send command," as set forth in Appellant's independent claim 1. In fact, nowhere in Osborne is there any specific disclosure regarding the arrow passing through Osborne's processor 54 to Osborne's network interface 84. As such, the Appellant maintains that there is no specific disclosure regarding anything being sent between Osborne's processor 54 and Osborne's network interface 84 (i.e., the mere illustration of an arrow passing through Osborne's processor 54 to Osborne's network interface 84 is not specific disclosure).

[16] Examiner's Answer, Page 30, Lines 4-11.

Instead, in reference to the corresponding section in Column 7, Lines 32-37, the arrow seems to merely relate to the messages passed by operating system 56 over network 82 via network interface 84 **during data transfer** (i.e., after any initiation process would already have been completed). In other words, nowhere in Osborne is there any disclosure of any initiation process of an RDMA operation occurring between Osborne's processor 54 and Osborne's network interface 84, let alone a "**one-shot initiation process comprising communicating a single command message comprising: buffer command information, and a write command to write a send command**," as set forth in Appellant's independent claim 1.

Further, the mere disclosure of various arrows between application 58 and application 78 does not teach an RDMA operation as alleged by the Examiner's Answer. In fact, nowhere in Osborne is there any disclosure of an RDMA operation. Put simply, "RDMA" and "remote direct memory access" are well-known terms readily used and understood by those of ordinary skill in the art. Further, the well-known terms "RDMA" and "remote direct memory access" do not appear in Osborne because Osborne does not teach RDMA operations.

Regarding the third argument in the Examiner's Answer, the Examiner's Answer states the following:

> As to the appellants' allegation that Osborne's "Send" command does not contain any buffer command information, the examiner would like to refer to the second parameter "Source Address" (which is a pointer to the

> message buffer) and the third parameter "Size" (that indicates the amount of data in the message buffer that is to be transported by the RDMA operation) described above. The appellants' argument is therefore without any merit.[17]

As acknowledged by the Examiner's Answer, Osborne's send command 67 merely includes information regarding an address and a size of message data in application memory 66, which themselves are not commands. As such, the Appellant maintains that Osborne's send command does not contain any buffer command information.[18]

Regarding the fourth argument in the Examiner's Answer, the Examiner's Answer states the following:

> The examiner cannot disagree more with the appellants' assertion that it is the message data copied/mapped from the application memory 66 to operating system message buffer 62 that is sent over network 82 via network interface 84, not the send command 67 sent between application 58 and operating system 56. As a person of ordinary skill in the art would know that in a network, data does not get transmitted without any command, destination address, and other applicable parameters. Every data packet has a header that includes a command followed by associated parameters. The NIC 84 translates the "Send" command shown in Fig. 1 to TCP-IP specific binary command that the other nodes in the network 82 understand and forward the message data across the network to the destination NIC 86.[19]

Regarding the Examiner's allegation that "[t]he examiner cannot disagree more with the appellants' assertion that it is the message data copied/mapped from the application memory 66 to operating system message buffer 62 that is sent over network 82 via

[17] Examiner's Answer, Page 30, Lines 12-17.
[18] Osborne, Column 7, Lines 22-25.
[19] Examiner's Answer, Page 30, Line 18 – Page 31, Line 5.

network interface 84, not the send command 67 sent between application 58 and operating system 56," the Appellant notes that Osborne explicitly states the following:

> First, the application 58 invokes a send command in step 99. An example way for an application 58 to invoke a send operation is by executing a command as depicted by the "send" command 67. **The command includes** an an identifier (ID) which is used in part to identify the receiver, **a source address from which message data will be taken and a size, indicating the amount of data to be sent. The operating system then copies the message in step 100 from application memory**, such as an endpoint 66, to message buffers, e.g., 62, in the operating system 56 of sender 50. This step is often optimized by mapping locations in the application memory to locations in the message buffer 62 to avoid actual copying. The operating system 56 then performs protocol processing if necessary in step 102. That is, the data to be sent is placed into the proper format as may be required by the network 82. **The message, or perhaps several messages if the amount of data is large, is then injected in step 104 into the network 82 through network interface 84.**

In other words, as the Appellant argued in the Brief on Appeal and as Osborne explicitly supports, it is the data copied/mapped from the application memory 66 to operating system message buffer 62 (i.e., the message) that is injected into the network 82 through network interface 84 (i.e., sent over network 82 via network interface 84), not the send command 67 sent between application 58 and operating system 56. As such, any allegation in the Examiner's Answer that suggests that Osborne's send command 67 is communicated any further than the operating system 56 of the sending node 50 is clearly unsupported by Osborne.

Regarding the allegations in the Examiner's Answer that the data transmitted over network 82 would include header data, the Appellant first notes that the header data does not include the address and size of message data in the application buffer 66 of the sending node 50 (i.e., information in the send command 67) because the data would already be copied/mapped to operating system message buffers 62. As such, one of ordinary skill in the art would instantly understand that the send command 67 is not a message header for data sent over network 82. Second, even if the send command 67 was a message header for data sent over network 82, the messages sent from operating system 56 over network 82 via network interface 84 is the data transfer operation (i.e., after any initiation would have been completed). As such, the passing of messages that include headers during data transfer from Osborne's operating system 56 into the network 82 through network interface 84 fails to provide any disclosure regarding an initiation process between a driver and a NIC of a host.

Regarding the above allegations in the Examiner's Answer that "[t]he NIC 84 translates the "Send" command shown in Fig. 1 to TCP-IP specific binary command that the other nodes in the network 82 understand and forward the message data across the network to the destination NIC 86," the Appellant notes that nowhere in Osborne is there any support for the Examiner's allegations. Instead, as discussed above, Osborne merely teaches sending the send command 67 between the application 58 and operating system 56. The data sent from the operating system 56 over the network 82 via the network interface 84 is message data identified by the send command, not the send command itself. As noted above, any allegation in the Examiner's Answer that suggests that Osborne's send command 67 is communicated any further than the operating system 56 of the sending node 50 is clearly unsupported by Osborne.

Therefore, the Appellant maintains that at least the limitations "wherein a **one-shot <u>initiation</u> process** of an RDMA operation is **<u>performed between the driver and the NIC</u> of the host, the one-shot initiation process comprising communicating <u>a single command message</u> comprising: buffer command information, <u>and</u> a write command to write a send command**," as set forth in Appellant's independent claim 1, are not anticipated by Osborne.

Accordingly, independent claim 1 is not anticipated by Osborne and is allowable. Furthermore, the Appellant reserves the right to argue additional reasons beyond those set forth herein to support the allowability of claim 1.

B. Rejection of Dependent Claims 2-5, 10 and 16

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 2:

> In response to the appellants argument that "copying a send command", as recited in claim 2, is different from "copying message data" disclosed by Osborne, the examiner states that the message data includes both the command an the data (payload). As explained earlier, every data packet must include a header with at least one command and any associated parameters, otherwise, payload by itself cannot be transported across the network.
>
> The appellants' allegation that Osborne clearly cannot disclose "wherein the driver posts the single command message to perform the one-shot initiation process", because nowhere in Osborne is there any disclosure

> regarding the functions of the processor 54, is without any merit, because the processor 54 interfaces with the NIC 84. As one of the interface functions, the processor 54 issues commands to NIC 84 to communicate with remote nodes. The "Send" command shown in Fig. 1 of Osborne, and disclosed in column 7, lines 15-45, is one such command that initiates transfer of message data to the remote node 52. The operating system 56, in the Osborne reference, provides machine-coded instructions that are executed by the hardware of the processor 54 to post a single "Send" command to NIC 84, as recited in claim 2 of the appellants.[20]

With regard to the above-arguments set forth in the Examiner's Answer, the Appellant notes that in the Brief on Appeal, the Appellant argued that the cited sections of Osborne failed to teach any functions of the processor 54, let alone disclosure regarding the processor 54 posting a single command message to perform a one-shot initiation process. Instead of citing to sections in Osborne that support the Examiner's allegations, the Examiner's Answer merely sets forth an imaginative (yet again, unsupported) interpretation of Osborne's disclosure. With regard to the anticipation rejections, MPEP 2131 states, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 2 USPQ2d 1051, 1053 (Fed.Cir. 1987). MPEP 2131 also states, "[t]he identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The only sections of Osborne cited by the Examiner in reference to the above arguments is Figure 1 and Column 7, Lines 15-45. Osborne's Figure 1 illustrates the following:

[20] Examiner's Answer, Page 31, Line 9 – Page 32, Line 3.

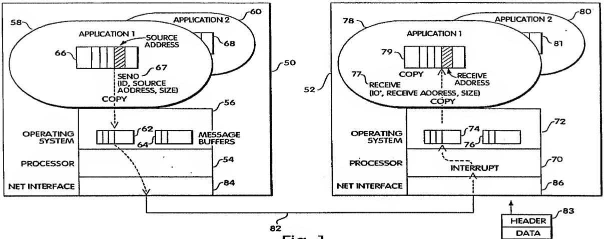

Fig. 1
(Prior Art)

Osborne's Column 7, Lines 15-45 recites the following:

> An application, such as 58, at the sender 50 and receiver 52 communicate by sending messages to each other across the network 82. The message may include any data, including a procedure call. As shown in FIG. 2, communication conventionally involves at least the following steps. First, the application 58 invokes a send command in step 99. An example way for an application 58 to invoke a send operation is by executing a command as depicted by the "send" command 67. The command includes an an identifier (ID) which is used in part to identify the receiver, a source address from which message data will be taken and a size, indicating the amount of data to be sent. The operating system then copies the message in step 100 from application memory, such as an endpoint 66, to message buffers, e.g., 62, in the operating system 56 of sender 50. This step is often optimized by mapping locations in the application memory to locations in the message buffer 62 to avoid actual copying. The operating system 56 then performs protocol processing if necessary in step 102. That is, the data to be sent is placed into the proper format as may be required by the network 82. The message, or perhaps several messages if the amount of data is large, is then injected in step 104 into the network 82 through network interface 84.
>
> Usually, message arrival at the receiver causes an interrupt to the processor 70, and the operating system 72 directly extracts the message in step 106 from the network (meaning network and network interface) and copies it into message buffer 74 in the operating system. Alternatively, in a

> system with appropriate hardware, the operating system may set up a direct memory access (DMA) with the network interface to extract and copy the message to message buffer 74.

As clearly shown above, nowhere in the cited sections of Osborne is there any disclosure regarding "wherein the driver posts the single command message to perform the one-shot initiation process," as set forth in Appellant's dependent claim 2. In fact, as noted in the Appellant's Brief on Appeal, nowhere in the cited sections of Osborne is there any disclosure of the function of processor 54. In fact, the only mention of processor 54 in the Examiner-cited sections of Osborne is the illustration in Figure 1. Because the Final Office Action and Examiner's Answer has failed to show "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference" as required for an anticipation rejection under MPEP 2131, the Appellant notes that the Examiner has failed to set forth a *prima facie* case of anticipation under 35 U.S.C. § 102(b).

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 3:

> The examiner further disagrees with the appellants' argument that "mapping locations in the application memory to message buffers does not pin-down memory buffers of the host", as recited in claim 3. When message buffers are allocated and mapped in the host memory as shown in Fig. 1, those memory locations are reserved for the duration of the operation (i.e., pinned down) until specifically released by the operating system. Thus Osborne shows and provides sufficient disclosure to teach claim 3 element, which recites that "the buffer command information

> comprises a command to describe pinned-down memory buffers of the host.[21]

However, nowhere in Osborne is there any support for the Examiner's allegation that mapping locations in the application memory to message buffers pins-down memory buffers for the duration of the operation until specifically released by the operating system. Further, even if the mapping of the message buffers by Osborne's operating system 56 resulted in the message buffers being pinned down as alleged in the Final Office Action and the Examiner's Answer (which is unsupported by Osborne), Osborne could not teach buffer command information comprises a command to describe pinned down memory buffers of the host. Put another way, if Osborne's mapping results in buffers being pinned down and Osborne's mapping occurs based on the send command 67, Osborne's send command 67 cannot comprise information describing buffers that have already been pinned down. Additionally, the send command 67 sent from an application 58 to an operating system 56 is not sent between a driver and a NIC of a host. As such, Osborne clearly fails to disclose "wherein the buffer command information comprises a command to describe pinned-down memory buffers of the host," as recited in Appellant's dependent claim 3.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 4:

[21] Examiner's Answer, Page 32, Lines 4-11.

> The appellants' assertion that Osborne does not disclose any steering tag is also without any merit. The steering tag/STag is used as a pointer that holds address of a buffer (please refer to appellants' provisional application 60/404,709 to page 3, paragraph 09, item b) which defines STag as "NIC builds a mapping table for the pinned down buffer on host 1. This mapping table is given a handle (i.e. a pointer) called an STag", which compares well with Osborne's recitation in column 7, lines 25-31 that teach "The operating system then copies the message from application memory to message buffers 62, in the operating system 56 of sender 50. This step is often optimized by mapping locations in the application memory to locations in the message buffer to avoid actual copying."). Osborne's "Send" message clearly includes such a pointer in the form of "Source Address" parameter that references (binds) a portion of the pinned-down (reserved) memory buffers of the host, as recited in appellants' claim 4.[22]

However, as discussed above, the Examiner alleges that Osborne's send command 67 sent from application 58 to operating system 56 directs the operating system to map locations in application memory 66 to message buffers 62 in the operating system 56. The Examiner further alleges that the mapping of the locations in application memory 66 to message buffers 62 in the operating system 56 results in memory buffers of the host being pinned-down (which the Appellant notes is unsupported by Osborne). Even if Osborne's mapping did result in memory buffers being pinned-down as alleged by the Examiner (which is unsupported by Osborne), the Appellant notes that such interpretation precludes Osborne from teaching "wherein the buffer command information comprises a command to bind a portion of the pinned-down memory buffers of the host to a steering tag (STag)," as recited in Appellant's dependent claim 4. Specifically, if the memory buffers are pinned down after the send command 67 is sent from application 58 and acted upon by operating system 56, the send command 67

[22] Examiner's Answer, Page 32, Line 12 – Page 33, Line 2.

cannot include a command to bind a portion of memory that has already been pinned-down to a steering tag (STag).

Basically, as one of ordinary skill in the art would readily understand, a Send command sent from an application to an operating system that merely includes a source address does not provide a command to bind a portion of the pinned-down memory buffers of the host to a steering tag (STag). In fact, nowhere in Osborne are steering tags disclosed. Instead, as noted by the Examiner, an STag may be a handle of a mapping table built by a NIC. Osborne fails to teach mapping tables built by NICs, let alone handles of such mapping tables, let alone commands to bind a portion of pinned-down memory buffers of the host to a handle of a mapping table built by a NIC. As such, Osborne clearly fails to disclose "wherein the buffer command information comprises a command to bind a portion of the pinned-down memory buffers of the host to a steering tag (STag)," as recited in Appellant's dependent claim 4.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 5:

> In response to the appellants' argument that the send message of Osborne, including a receiver ID, source address and size of message data does not teach an RDMA send message, as disclosed in claim 5, the examiner would like to repeat that a person of ordinary skill in the art would readily be able to understand, by looking at Fig. 1 of Osborne and reviewing the disclosure in column 7, lines 15-45 that the send command is indeed an RDMA send message.[23]

[23] Examiner's Answer, Page 33, Lines 3-8.

However, as noted in the Appellant's Brief on Appeal, one of ordinary skill in the art would readily be able to understand that the mere disclosure of a send message including a receiver ID and a source address and size of message data does not teach an RDMA send message. Further, Osborne's teaching that DMA can be used in the receiver does not teach that a send message in a sender is an RDMA send message. In fact, nowhere in Osborne is there any mention of "remote direct memory access" or "RDMA." As such, Osborne clearly fails to disclose "wherein the send command is an RDMA send message," as recited in Appellant's dependent claim 5.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 16:

> The examiner disagrees with the appellants' argument that performing DMA operations across network do no correspond to RDMA operations across a network, as recited in appellants' dependent claim 16. Fig. 1 clearly shown and column 7, lines 15-45 disclose that nods 50 and 52 are remote from each other, connected via network 82, and perform a DMA (Direct Memory Access) operation between the two nodes 50 and 52. Therefore, it does not matter that nowhere in Osborne is there any mention of "remote direct memory access" or "RDMA", so long as one of the ordinary skill in the art can conclude that Osborne does teach such an operation.[24]

However, the Examiner's Answer mischaracterizes the Appellant's arguments in the Brief on Appeal and the disclosure of Osborne. Specifically, the Appellant did not argue that Osborne's disclosure of performing a DMA operation across a network does not disclose RDMA. Instead, the Appellant argued that Osborne does not teach performing

[24] Examiner's Answer, Page 32, Line 12 – Page 33, Line 2.

a DMA operation across a network and instead performs the DMA operation between a network interface 86 and operating system 72 at a receiving node 52. As one of ordinary skill in the art would readily understand, performing a DMA operation between a network interface 86 and operating system 72 at a receiving node 52 as disclosed by Osborne at Column 7, Lines 38-45 fails to teach performing a DMA operation across a network as alleged by the Examiner. Also, nowhere in Osborne is there any mention of "remote direct memory access" or "RDMA." As such, Osborne clearly fails to disclose "wherein the NIC comprises an RDMA-enabled NIC," as recited in Appellant's dependent claim 16.

Accordingly, the Appellant maintains that claims 2-5, 10 and 16 are allowable over the reference cited in the Final Office Action at least for the above reasons and the reasons set forth in the Appellant's Brief on Appeal. The Appellant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 2-5, 10 and 16.

II. Claims 6-9, 11 and 24 Are Not Obvious Over Osborne in view of Roach

Claims 6-9, 11 and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Osborne in view of Roach. The Appellant stands by the arguments made in the corresponding sections of the Brief on Appeal as set forth in further detail below.

A. Rejection of Independent Claim 24

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer sets forth three arguments in response to the Appellant's arguments in the corresponding section of the Brief on Appeal.

Regarding the first argument in the Examiner's Answer, the Examiner's Answer states the following:

> On pages 17-21 of appeal brief, the appellants repeat the same arguments for the method claim 24 that were presented for the system claim 1 in section I-A. The examiner has already responded to each repeated argument in the "Examiner's Response" section, Ground 1-A for claim 1. The same response is applicable to claim 24.[25]

As such, for at least the reasons stated previously with regard to claim 1, the Appellant submits that claim 24 is allowable over the combination of Osborne and Roach, as well. Additionally, the Appellant submits that claim 24 is independently allowable as discussed in further detail in the Appellant's Brief on Appeal and below.

Regarding the second argument in the Examiner's Answer, the Examiner's Answer states the following:

> In response to the appellants' argument that the cited reference of Roach et al. does not describe any STag values, the examiner would like to state

[25] Examiner's Answer, Page 34, Lines 4-8.

> that STag is a term used by the appellants for a pointer to locate data stored at specified (value of the pointer) memory location (please see Examiner's Response for Ground 1-B). As can be clearly seen in Figs. 8-9 of Roach et al., a series of pointers (BSWA or Buffer Staring Word Address representing STag value) are used as buffer list entry pointers. In Fig. 8, Roach further shows the bit-level details of each such pointer entry (two 32-bit words), wherein the first word (word 0) is the pointer that holds the address of a buffer storing message data, and the second word (word 1) stores control information including BLM (Buffer List Modifiers) field, the bit-level details of which are listed in column 8, lines 44-67 through column 9, lines 1-18. Some of these bits (e.g., bits 31, 28-25) are used for buffer pointer validation purposes.[26]

However, as acknowledged by the Examiner in the Examiner's Response for Ground 1-B, an STag value is not merely any pointer. Instead, the section of the Appellant's provisional application cited for the definition of STag disclosed that an STag is a handle for a mapping table built by a NIC. As such, Roach's BSWA pointers are clearly not STag values as defined in the Appellant's provisional application referred to by the Examiner. Instead, Roach merely teaches placing predetermined bits indicating which is the last frame in a series of frames in a Fibre Channel frame header (FC-2 header).[27] As such, Roach clearly fails to teach STag values.

Regarding the third argument in the Examiner's Answer, the Examiner's Answer states the following:

> The examiner disagrees with appellants' argument that Roach et al. do not teach RDMA operation. Roach does teach DMA operation (for example, column 4, lines 53-63 and several other paragraphs). Sine these operations involve data transfer between network nodes separated across

[26] Examiner's Answer, Page 35, Lines 4-15.

[27] *See e.g.*, Roach, Abstract, Column 6, Lines 21-25 and Column 8, Lines 54-55.

> a network, RDMA operation is taught in the Roach et al. reference, even if not explicitly mentioned as RDMA. In any case, Osborne reference already teaches RDMA. Roach et al. reference was combined with the teachings of Osborne to describe features corresponding to "inserting STag value in a first field of a DDP or RDMA header of the RDMA send message," and validating the STag value in the first field with a bit flag or other specified value in a second field (e.g. BLM) of the DDP or RDMA header". Even if Roach used FC-2 header (Fiber Channel frame header) to describe the insertion of STag pointers and setting BLM bits to validate the STag pointers, the concept may be applied to an RDMA header as well.[28]

However, Roach's Column 4, Lines 53-63 merely describe a DMA operation between a full duplex communication processor 22 and a host memory 42 at the same node 20 as illustrated in Roach's Figure 3. Nowhere in Roach is there any disclosure regarding performing DMA operations across a network. As discussed above, Osborne also fails to teach performing RDMA operations.

Further, with regard to the cited sections of Roach, Roach explicitly teaches building a Fibre Channel frame header (FC-2 header)[29] and the Examiner has failed to provide any motivation or support in the cited references for using DDP or RDMA headers. In fact, one of ordinary skill in the art would readily understand that the concepts applied to a Fibre Channel frame header (FC-2 header) are inoperable when applied to the completely different DDP and RDMA headers.

Clearly, Osborne merely teaches communicating a message from a sender 50 to a receiver 52 without disclosing any initiation process of an RDMA operation performed

[28] Examiner's Answer, Page 35, Lines 4-15.
[29] Roach, Column 8, Lines 54-55.

between the driver and the NIC of the host, let alone a one-shot initiation process,[30] and Roach merely teaches placing predetermined bits indicating which is the last frame in a series of frames in a Fibre Channel frame header (FC-2 header).[31]

Therefore, the Appellant maintains that at least the limitations of "initiating an RDMA write operation using **a one-shot initiation process between a driver and a NIC** of a host, wherein **the one-shot initiation process comprises communicating a single command message comprising: buffer command information comprising commands to insert and validate an STag value, and a write command to write an RDMA send message,**" "inserting the **STag value** in a first field of a **DDP or RDMA header** of the RDMA send message," and "validating the **STag value** in the first field with a bit flag or other specified value in a second field of the **DDP or RDMA header**," as set forth in Appellant's independent claim 24, are not unpatentable over the combination of Osborne and Roach.

Accordingly, independent claim 24 is not unpatentable over Osborne in view of Roach and is allowable. Furthermore, the Appellant reserves the right to argue additional reasons beyond those set forth herein to support the allowability of claim 24.

[30] *See e.g.*, Osborne, Abstract; Column 4, Lines 62-64; Column 5, Lines 14 and 24-30; Column 8, Lines 16-19, 27-33 and 63-66; Column 9, Lines 1-4, 10-11 and 33-43; and Column 10, Lines 6-7.
[31] *See e.g.*, Roach, Abstract, Column 6, Lines 21-25 and Column 8, Lines 54-55.

B. Rejection of Dependent Claims 6-9 and 11

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claims 6-9:

> The examiner contends that Roach et al. show and disclose all the claim elements of dependent system claims 6-9 in Figs. 8-9 and column 8, lines 32-67 through column 9, lines 1-18. For example, Fig. 9 shows a fiber channel header with reserved fields and optional fields as well as undefined fields, wherein the optional fields are used to place BSWA pointer (STag) values in them (recited in claim 6), and to encode (recited in claim 7) and set (recited in claim 8) the bit values in the BLM field to indicate validity of the BSWA pointers; and by setting one or more bits or encoding a value for the set bits, advertises (recited in claim 9) the corresponding buffers addressed by the BSWA pointers to be valid (e.g., Bit 31 when set to 1, indicates receive only buffer, but when set to a '0'b, may be used for transmit buffers). Fig. 3 shows a host 40 interface with a communication processor 22 that incorporates receive and transmit protocol engines 30 and 32 respectively and NL-Port 36 (all part of NIC functions); column 4, lines 53-63 disclose the same details.[32]

However, as discussed above with regard to Appellant's independent claim 24, the Appellant notes that Roach merely teaches placing predetermined bits indicating which is the last frame in a series of frames in a Fibre Channel frame header (FC-2 header).[33] Nowhere in Roach is there any disclosure of STag values. As acknowledged by the Examiner in the Examiner's Response for Ground 1-B, an STag value is not merely any pointer. Instead, the section of the Appellant's provisional application cited for the definition of STag disclosed that an STag is a handle for a mapping table built by a NIC.

[32] Examiner's Answer, Page 36, Lines 9-21.
[33] *See e.g.*, Roach, Abstract, Column 6, Lines 21-25 and Column 8, Lines 54-55.

As such, Roach's BSWA pointers are clearly not STag values as defined in the Appellant's provisional application referred to by the Examiner. Thus, Roach clearly fails to teach STag values.

Further, with regard to the cited sections of Roach, Roach explicitly teaches building a Fibre Channel frame header (FC-2 header).[34] One of ordinary skill in the art would readily understand that the concepts applied to a Fibre Channel frame header (FC-2 header) are inoperable when applied to the completely different DDP and RDMA headers. As such, even if the information inserted into Roach's Fibre Channel frame header (FC-2 header) included STag values (which is clearly not disclosed by Roach), Roach would still fail to teach "wherein the NIC places the STag value in an optional field **in a direct data placement DDP or RDMA header**," as recited in Appellant's dependent claim 6; "wherein the NIC encodes a value into a field **in the DDP or RDMA header** indicating that the STag value in the optional field is valid," as set forth in the Appellant's dependent claim 7; "wherein the NIC sets one or more bits in a field **in the DDP or RDMA header** indicating that the STag value in the optional field is valid" as recited in Appellant's dependent claim 8; and, "wherein the NIC sets one or more bits or encodes a value into a second field **in the DDP or RDMA header** to advertise the portion of the pinned memory buffers of the host associated with the STag," as set forth in the Appellant's dependent claim 9.

[34] Roach, Column 8, Lines 54-55.

Accordingly, the Appellant maintains that claims 6-9 and 11 are allowable over the references cited in the Final Office Action at least for the above reasons and the reasons set forth in the Appellant's Brief on Appeal. The Appellant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 6-9 and 11.

III. Claims 12-15 Are Not Obvious Over Osborne in view of Roach and further in view of Tillier

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claims 12:

> The examiner has already responded to the appellants' argument about the use of the term "steering tag/STag" in Ground 1-B. As to the appellants' argument that Tillier does not allocate the pointer because Tillier's I/O unit receives the pointer from the host, the examiner would like to state that the Tiller reference was added to merely show and disclose a device driver 103-2 (see Fig. 1), because the cited Osborne reference does not specifically refer to the operating system 56 modules and processor 54 as driver recited in claim 12. Otherwise, Osborne clearly shows allocation of a pointer (column 7, lines 25-31) that corresponds to the "source address" parameter of the "Send" command shown in Fig. 1. Furthermore, the argument that Tillier reference only receives but does not allocate an STag is also incorrect. As shown in Fig. 8 of Tillier, there are two different pointers, one received from the Host (to the left in Fig. 8) along with the command, and the other (Pointer to "A") allocated by the I/O unit to store Data "A" at a location in the storage device specified by the Pointer to "A".[35]

[35] Examiner's Answer, Page 37, Line 18 – Page 38, Line 10.

However, regarding the Examiner's argument in connection with the term "STag," the Appellant notes that the Examiner's comments in Ground 1-B merely illustrate that the mere disclosure of pointers does not teach STags. As such, the combination of references cited by the Examiner fails to teach the STag limitations.

Regarding the Examiner's comment that Osborne fails to teach a driver, the Appellant notes that such admission is further evidence of the improper rejection of Appellant's independent claim 1 (which recites a driver) under 35 U.S.C. §102(b). Further, because Tillier fails to teach a driver of a host, Tillier fails to remedy the deficiencies of Osborne.

Regarding the Examiner's comment that Tillier's Figure 8 illustrates two different pointers, one received from the Host (to the left in Fig. 8) along with the command, and the other (Pointer to "A") allocated by the I/O unit to store data "A" at a location in storage device specified by the Pointer to "A," the Appellant notes that the Pointer to "A" is the same pointer as the "Command & Pointer" provided by the Host to the I/O unit.[36]

Put simply, one of ordinary skill in the art would recognize that Tillier's mere disclosure of a "Pointer to 'A'" fails to disclose an STag. In fact, nowhere in Tillier is there any mention of the terms "steering tag" and/or "STag," which are well known terms in the art. Further, Tillier's Figure 8 clearly shows that the I/O unit is separate from the host. As such, Tillier's I/O units is not a driver of a host. Additionally, as also shown in Tillier's Figure 8, the "Pointer to 'A'" is provided by the host to the I/O unit (i.e.,

[36] *See e.g.*, Tillier, Figure 8 and Column 1, Lines 25-46.

"Command & Pointer"). As such, even if Tillier's I/O unit was a driver of a host (which it is not), and even if Tillier's "Pointer to 'A'" could be considered an STag (which it is not), Tillier's I/O unit clearly does not allocate the pointer because Tillier's I/O unit receives the pointer from the host.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 13:

> The examiner begs to differ with the appellants' assertion (for claim 13) that STag values cannot be incremented. As used by the appellants' STag represents a pointer, whose value refers to an address of a buffer or memory location. Such pointer values are often incremented or decremented to point to new locations in memory. The reason Roach increases the put pointer is because a new command stored in the command ring buffer must be stored at a different address than the previous command's location in the buffer. Since the appellants' invention uses only a single command, there is not need to increase the pointer value. But, that does not mean the STag values cannot be incremented in any situation, as the appellants assert.[37]

However, the Appellant first notes that the Examiner mischaracterizes the term "STag" as is known in the art and used in the Appellant's specification. Specifically, an STag is not merely any pointer as alleged by the Examiner. Nor does an STag merely point to an address of a buffer or memory location as alleged by the Examiner. Instead, as acknowledged by the Examiner in the Examiner's Response for Ground 1-B, the section of the Appellant's provisional application cited by the Examiner for the definition of STag discloses that an STag is a handle for a mapping table built by a NIC. Handles for

[37] Examiner's Answer, Page 38, Lines 11-19.

mapping tables are not incremented. As such, Roach's put and get pointers, which are incremented as commands and added to and removed from the command ring, cannot be STag values.

Accordingly, the Appellant maintains that claims 12-15 are allowable over the reference cited in the Final Office Action at least for the above reasons and the reasons set forth in the Appellant's Brief on Appeal. The Appellant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 12-15.

IV. Claim 17 Is Not Obvious Over Osborne in view of Pandya

Claim 17 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Osborne in view of Pandya. The Appellant stands by the arguments made in the corresponding sections of the Brief on Appeal as set forth in further detail below.

A. Rejection of Independent Claim 17

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer sets forth four arguments in response to the Appellant's arguments in the corresponding section of the Brief on Appeal. As noted below the Examiner's Answer fails to respond to the Appellant's argument at Page 30, Line 18 through Page 31, Line 12.

Regarding the first argument in the Examiner's Answer, the Examiner's Answer states the following:

> The examiner would like to refer to Fig. 33 in the Pandya reference, because the Pandya invention is titled "TCP/IP Processor and Engine using RDMA". Therefore, the entire invention uses RDMA and either of the two figures (33 or 35) will suffice. The examiner chose Fig. 33, because it is simpler in operation.[38]

However, the Appellant notes that despite Pandya's invention title, Pandya's Figure 33 is entitled "Write Operation" and as shown at least at blocks 3302 and 3308, the write operation is using SCSI, not RDMA. For further support, the Appellant recommends viewing Pandya's Figure 31, entitled "Read Operation" and is described at Column 31, Lines 38-40 as "[t]his transaction is a standard SCSI READ transaction with the data transport over IP based storage protocol like iSCSI as the PDUs of that protocol." Put simply, despite Pandya's title, Pandya's SCSI READ operation in Figure 31 is not an RDMA operation. Similarly, despite Pandya's title, Pandya's SCSI WRITE operation in Figure 33 is also not an RDMA operation. As such, the Examiner can refer to Figure 33; however, in doing so, the Examiner will not be referring to any RDMA operations.

Regarding the second argument in the Examiner's Answer, the Examiner's Answer states the following:

[38] Examiner's Answer, Page 40, Lines 1-4.

> As to appellants' argument that in Pandya the buffers are not freed until the status of the command execution is passed onto the host SCSI layer, which releases the buffers, which is different than the claimed feature of "completion message comprising a send completion indication, and buffer freeing status information", the examiner would like to clarify that once the receiving node has copied all the transmitted message data successfully, it frees its own buffers and sends a normal completion status (such as a return code set to zero) and suggesting to the sending node to free its own buffers. The receiving node cannot free the allocated buffers (resources) of the sending node, since it has no such access rights, it can only indicate the status of its own buffers that have been freed, after the normal transfer of message data.[39]

However, the Appellant first respectfully notes that the Examiner is not clarifying his position, but changing his position. For example, as shown at Page 17, Lines 13-20 of the Examiner's Answer and at Page 17, Lines 12-22 of the Final Office Action, the Examiner previously argued that Pandya's disclosure of freeing resources at the host SCSI layer (i.e., after passing the completion message without buffer freeing status information between a driver and a NIC of the host) taught the Appellant's claim limitations. After realizing that the cited-section of Pandya fails to teach the Appellant's claim limitations, the Examiner now presents the new position discussed above.

The Examiner's new position outlined above (which fails to cite to any of the text or figures of Pandya) alleges that Pandya teaches that its "Send Status & Sense" sent from the target to the initiator includes information indicating that the target's buffers have been freed. The Appellant notes that nowhere in Pandya is there any disclosure regarding including information indicating that the target's buffers have been freed in the "Send Status & Sense" message sent from the target to the initiator. Further, as one of

[39] Examiner's Answer, Page 40, Lines 4-14.

ordinary skill in the art would readily understand, the initiator has no interest in the buffer freeing status of the target at the completion of an RDMA operation. As such, as is known in the art, target buffer free status information is not sent to an initiator at the completion of an RDMA transfer. As such, the Examiner's new position is neither supported by Pandya, nor logical in view of that which is well known in the art.

Regarding the third argument in the Examiner's Answer, the Examiner's Answer states the following:

> The examiner also disagrees with the appellants' assertion that "as is well known in the art, sense status is merely error information" that cannot represent "send completion indication" and "buffer freeing error information" features of claim 17. What is well known in the art is that a return code with a zero value generally represents normal completion of an operation, while a non-zero value represents a code in a look-up table for a plurality of abnormal ending (abend) operations.[40]

The argument referred to by the Examiner states that "Pandya merely teaches sending a command completion and sense status once all the data has been transferred. As is well known in the art, sense status is merely error information, which is different than buffer freeing status information (i.e., information indicating that the buffers have been freed)." The Appellant notes that the Appellant's argument is accurate. As acknowledged by the Examiner, sense status merely indicates whether an error occurred and if an error occurred, identifies the error that occurred (i.e., error information). As such, the mere identification of whether an error occurred and if an

[40] Examiner's Answer, Page 40, Lines 14-20.

error occurred, identifying the error that occurred does not teach buffer freeing status information (i.e., information indicating that the buffers have been freed). Put simply, there is no sense code that indicates buffer freeing status information of the target. Instead, the sense status provides information regarding the data transmission. As such, Pandya's mere disclosure of sending a completion message including sense status fails to teach "wherein a **one-shot completion process** of an RDMA operation is **performed between the driver and the NIC of the host, the one-shot completion process comprising communicating a single completion message comprising: a send complete indication, and buffer freeing status information**," as recited in Appellant's independent claim 17.

The Appellant notes that the Examiner fails to address the Appellant's argument at Page 30, Line 18 through Page 31, Line 12. Specifically, the Appellant's Brief on Appeal noted that the Final Office Action alleges that Osborne's disclosure of extracting a message from the network into system message buffers at a receiver, protocol processing, and copying the message data from the system message buffers to application memory, teaches a completion process. However, as stated in MPEP 2106(II)(C), "when evaluating the scope of a claim, every limitation in the claim must be considered. USPTO personnel may not dissect a claimed invention into discrete elements and then evaluate the elements in isolation. Instead, the claim as a whole must be considered. See, e.g., *Diamond v. Diehr*, 450 U.S. 175, 188-89, 209 USPQ 1, 9

(1981)." As such, reviewing the Appellant's independent claim 17 as a whole, it is clear that the one-shot completion process occurs after the RDMA operation has been completed because the one-shot completion process comprises communicating a single completion message comprising a send complete indication (i.e., transfer is complete), and buffer freeing status information (i.e., indication that buffers have been freed). Thus, Osborne's disclosure regarding receiving message data at a receiver and transferring the message data to application memory is wholly unrelated to a one-shot completion process of an RDMA operation.

Regarding the fourth argument in the Examiner's Answer, the Examiner's Answer states the following:

> In response to the appellants' statement that Osborne's teachings cannot be combined with Pandya's teachings, the examiner would like to state that in a distributed data processing environment involving processes that execute at different network nodes, the combined teachings of Osborne and Pandya are often used as an inter-process communication stage at the end of an operation.[41]

The Appellant notes that nowhere in the combination of Osborne and Pandya is there any support for the Examiner's allegation. Instead, one of ordinary skill in the art would clearly not be motivated to combine Osborne's teaching regarding transferring data (i.e., **during** transfer **at a receiver**) with Pandya's teachings regarding forwarding a completion message received from a receiver **at the sender** to a driver at the sender **after a data transfer is complete**. Put another way, there is no suggestion to one of

[41] Examiner's Answer, Page 40, Lines 14-20.

ordinary skill in the art to combine the cited teachings of Osborne and Pandya because the relied on teachings are wholly unrelated (i.e., Osborne's teachings **during** data transfer **at a receiver** *versus* Pandya's teachings **after** data transfer **at a sender**).

Clearly, Osborne merely teaches communicating a message from a sender 50 to a receiver 52 without disclosing any completion process of an RDMA operation performed between the driver and the NIC of the host, let alone a one-shot completion process,[42] and Pandya merely teaches communicating a message indicating that the communication session is complete up to an SCSI layer, which releases the buffers used for the data transfer.[43]

Therefore, the Appellant maintains that at least the limitations of "wherein a **one-shot completion process** of an RDMA operation is **performed between the driver and the NIC of the host**, **the one-shot completion process comprising communicating a single completion message comprising: a send complete indication, and buffer freeing status information**," as set forth in Appellant's independent claim 17, are not unpatentable over Osborne in view of Pandya.

Accordingly, independent claim 17 is not unpatentable over Osborne in view of Pandya and is allowable. Furthermore, the Appellant reserves the right to argue additional reasons beyond those set forth herein to support the allowability of claim 17.

[42] *See e.g.*, Osborne, Abstract; Column 4, Lines 62-64; Column 5, Lines 14 and 24-30; Column 8, Lines 16-19, 27-33 and 63-66; Column 9, Lines 1-4, 10-11 and 33-43; and Column 10, Lines 6-7.
[43] Pandya, Column 13, Lines 35-48.

V. Claims 18, 20 and 22-23 Are Not Obvious Over Osborne in view of Pandya and further in view of Tillier

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 18:

> To summarize, as explained above, Osborne reference teaches a "source address" pointer parameter corresponding to STag value of claim 18, and since an allocated pointer represents address of a memory buffer reserved (i.e. pinned down) for an intended operation, and furthermore, since "source address" is not commonly used and defined parameter for a generic packet header, it can only be placed in the optional field area of the packet header. Therefore, Osborne does teach the elements of claim 18 at the sending node, whereas Tillier shows and discloses the corresponding pointer (STag) at the receiving node.[44]

However, as discussed above, nowhere in Osborne are steering tags disclosed. Instead, as noted by the Examiner, an STag may be a handle of a mapping table built by a NIC. Osborne fails to teach mapping tables built by NICs, let alone handles of such mapping tables, let alone the NIC receiving a message comprising an optional field carrying a handle of a mapping table built by the NIC.

Additionally, as noted above, Osborne's source address is not sent to Osborne's network interface 84. Therefore, even if Osborne's source address could be considered

[44] Examiner's Answer, Page 42, Lines 5-12.

an STag value (which it clearly is not), Osborne's source address is never received at a NIC. Further, even if Osborne's source address could be considered an STag value (which it clearly is not), Osborne's source address is not an STag value being associated with pinned memory **in a remote host**. Instead, Osborne's source address merely describes the location of the message in application memory 66 for copying or mapping to operating system buffers 62.

Tillier fails to remedy the deficiencies of Osborne. For example, the Final Office Action alleges that Tillier's disclosure of "Pointer to 'A'" in Figure 8 teaches an STag value associated with pinned memory in a remote host.[45] However, one of ordinary skill in the art would recognize that Tillier's mere disclosure of a "Pointer to 'A'" fails to teach an STag. In fact, nowhere in Tillier is there any mention of the terms "steering tag" and/or "STag," which are well known terms in the art. Further, Tillier's Figure 8 clearly shows that I/O unit is separate from Host.[46] As such, Tillier's I/O unit is not a NIC of a host. Additionally, nothing in Tillier teaches that the data stored at the host is pinned down. As such, Tillier fails to remedy the deficiencies of Osborne in view of Pandya in that the combination of references clearly fail to teach "wherein the NIC receives a message comprising an optional field carrying a STag value, the STag value being associated with pinned memory in a remote host," as recited in Appellant's dependent claim 18.

[45] Final Office Action, Page 18, Lines 10-19.
[46] Tillier, Figure 8.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 20:

> In response to appellants' argument that Tillier does not show or disclose freeing up reserved resources (buffers) after the transfer of message data from the sending host to the receiving host is complete, as disclosed in dependent claim 20, the examiner would like to state that Tillier's Fig. 6, steps 604-606 and column 8, lines 9-15 clearly show and disclose the claimed features by stating that "before terminating, the buffers mapped in the system are unmapped (i.e., de-associated from the pointer) in step 604, the I/O request is freed in step 605, and the resources associated with the I/O request are freed. Fig. 7 shows corresponding pseudo code for the disclosed features.[47]

However, nowhere in Tillier is there any mention of the terms "steering tag" and/or "STag," which are well known terms in the art. Further, even if STag values were disclosed (which is clearly not the case), nowhere in Tillier is there any teaching that the freeing of resources includes de-associating STag values from pinned-memory in the host. In fact, nothing in Tillier teaches that the data stored at the host is pinned down. Further, even if Tillier did disclose STag values (which Tillier does not), did disclose pinned-down memory (which Tillier does not) and did disclose de-associating STag values from the pinned-down memory in the host (which Tillier does not), Tillier still fails to teach that a NIC performs the de-association. As such, Tillier fails to remedy the deficiencies of Osborne in view of Pandya in that the combination of references clearly fails to teach "wherein the NIC de-associates the STag value with the pinned memory in the host, thereby preventing further access to the pinned memory using the de-associated STag value," as recited in Appellant's dependent claim 20.

[47] Examiner's Answer, Page 42, Lines 13-20.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 20:

> As to the argument that Tillier fails to teach that a NIC performs de-association of STag, as recited in claim 20, the examiner would like to state that this feature is already disclosed in the cited Pandya reference (see Fig. 7 for the architecture of RDMA operations that shows NICs both at the sending and receiving nodes, which when associated with the operations depicted in Fig. 33 and disclosed in column 13, lines 35-48, show that at the completion stage, the NICs free up the allocated buffers 705 and 710 at their respective nodes and the hosts free up their corresponding buffers 701 and 706.[48]

However, Pandya's disclosure at Figure 7 related to RDMA is different than Pandya's disclosure at Column 13, Lines 35-48 related to an iSCSI data flow and the SCSI Write operation illustrated at Pandya's Figure 33. Additionally, nowhere in Pandya is there any disclosure that the NIC is responsible for freeing buffers. Additionally, even if Pandya's NIC did free buffers (which is not disclosed by Pandya), nothing in Pandya teaches de-associating STag values with the pinned memory in the host. In fact, nowhere in Pandya is there any disclosure of STag values and pinned memory. As such, Pandya fails to remedy the deficiencies of Osborne in view of Tillier in that the combination of references clearly fails to teach "wherein the NIC de-associates the STag value with the pinned memory in the host, thereby preventing further access to the pinned memory using the de-associated STag value," as recited in Appellant's dependent claim 20.

[48] Examiner's Answer, Page 43, Lines 1-8.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 22:

> The appellants' argument that Osborne, in view of Pandya and Tillier, fails to disclose that the freeing of resources includes de-associating STag values from SGL information, as recited in claim 22, is without any merit, because Osborne shows (in Fig. 1) a plurality of message buffers that can be referenced by a list of pointers (corresponding to Scatter Gather List or SGL when the message buffers are not in contiguous memory space); and column 7, lines 28-31 that disclose the mapping of the scattered message buffer addresses; and the Pandya as well as Tillier references that describe freeing up the reserved buffers (i.e. de-associating STag values from SGL information) after the message data transfer has been transferred successfully, as listed above.[49]

However, the Appellant notes that nowhere in the cited sections of Osborne, Pandya and Tillier is there any disclosure regarding STag values, let alone an STag value associated with SGL information, let alone a NIC de-associating an STag value with previously associated SGL information. As such, the combination of Osborne in view of Pandya and further in view of Tillier clearly fails to teach "wherein the NIC de-associates the STag value with previously associated SGL information," as recited in Appellant's dependent claim 22.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 23:

[49] Examiner's Answer, Page 43, Lines 9-18.

> In summary, the response to arguments for dependent claim 23 is that the pinned down memory is no more than a temporarily reserved memory buffer that is for the exclusive use of the operation for which it is reserved, until freed.[50]

However, nowhere in Tillier is there any teaching that the freeing of resources includes freeing resources dedicated to information regarding the pinned-memory. As such, even if Tillier did disclose pinned memory (which Tillier does not), Tillier still fails to disclose freeing resources dedicated to information regarding the pinned memory. Further, as discussed above Tillier also fails to teach that a NIC frees resources. Osborne and Pandya fail to remedy the deficiencies of Tillier. As such, the combination of references clearly fails to teach "wherein the NIC frees any resources dedicated to information regarding the pinned memory," as recited in Appellant's dependent claim 23.

Accordingly, the Appellant maintains that claims 18, 20 and 22-23 are allowable over the reference cited in the Final Office Action at least for the above reasons and the reasons set forth in the Appellant's Brief on Appeal. The Appellant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 18, 20 and 22-23.

[50] Examiner's Answer, Page 43, Lines 19-21.

VI. Claim 19 Is Not Obvious Over Osborne in view of Pandya, in further view of Tillier, and still further in view of Roach

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Accordingly, the Appellant maintains that claim 19 is allowable over the reference cited in the Final Office Action at least for the above reasons and the reasons set forth in the Appellant's Brief on Appeal. The Appellant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claim 19.

VII. Claim 21 Is Not Obvious Over Osborne in view of Pandya, in further view of Tillier, and still further in view of Futral

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer states the following with regard to Appellant's dependent claim 21:

> For the response to appellants' argument on page 37 of the appeal brief, please refer to the examiner's response in Ground 1-B for claim 4.
>
> In response to the second argument on page 37 that Futral fails to teach any comparison by the driver, the examiner would like to refer to Fig. 4 in Futral, wherein two block of data are shown. The first block has a chain pointer 64 that point to (i.e. contains address of) the beginning of the second block. If there were additional blocks containing transaction detail elements (e.g., 4 total blocks instead of 2), the driver must compare the

> chain pointer address value with the other three blocks' starting address to maintain the order within the SGL list.[51]

However, regarding the Examiner's response in Ground 1-B for claim 4, as noted by the Examiner, an STag may be a handle of a mapping table built by a NIC. Futral fails to teach mapping tables built by NICs, let alone handles of such mapping tables, let alone a driver comparing a handle of mapping table to a handle of a mapping table previously sent.

Further, Futral's Figure 4 merely illustrates "a diagram of how various element types may be arranged to form the logical structure of an SGL."[52] Even if Futral's chain pointers 64 could be considered STags (which they are clearly not), nowhere in Futral's Figure 4 and supporting disclosure is there any teachings related to the comparison of the chain pointers 64, let alone the comparison of a received chain pointer value with a chain pointer value previously sent. In fact, nowhere in Futral is there any teaching of sending chain pointers.

As such, Futral fails to remedy the deficiencies of Osborne, Pandya and Tillier in that the combination of references clearly fails to disclose, at least, for example, "wherein the driver compares the STag value received with a STag value previously sent," as recited in Appellant's dependent claim 21.

Accordingly, the Appellant maintains that claim 21 is allowable over the reference cited in the Final Office Action at least for the above reasons and the reasons set forth

[51] Examiner's Answer, Page 45, Lines 1-9.
[52] See e.g., Futral, Column 2, Lines 41-43 and Column 6, Lines 23-24).

in the Appellant's Brief on Appeal. The Appellant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claim 21.

VIII. Claim 25 Is Not Obvious Over Osborne in view of Pandya and further in view of Roach

Claim 25 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Osborne in view of Pandya and further in view of Roach. The Appellant stands by the arguments made in the corresponding sections of the Brief on Appeal as set forth in further detail below.

A. Rejection of Independent Claim 25

The Appellant stands by the arguments made in the corresponding section of the Brief on Appeal.

Additionally, the Examiner's Answer sets forth four arguments in response to the Appellant's arguments in the corresponding section of the Brief on Appeal. Further, the Examiner's Answer fails to respond to the Appellant's argument at Page 43, Lines 1-16

Specifically, the Final Office Action alleges that Osborne's disclosure of extracting a message from the network into system message buffers at a receiver, protocol processing, and copying the message data from the system message buffers to application memory, teaches a completion process. However, as stated in MPEP

2106(II)(C), "when evaluating the scope of a claim, every limitation in the claim must be considered. USPTO personnel may not dissect a claimed invention into discrete elements and then evaluate the elements in isolation. Instead, the claim as a whole must be considered. See, e.g., *Diamond v. Diehr*, 450 U.S. 175, 188-89, 209 USPQ 1, 9 (1981) ." As such, reviewing the Appellant's independent claim 25 as a whole, it is clear that the one-shot completion process occurs after the RDMA operation has been completed because the one-shot completion process comprises communicating a single completion message comprising a send complete indication (i.e., transfer is complete), and buffer freeing status information (i.e., indication that buffers have been freed). Thus, Osborne's disclosure regarding receiving message data at a receiver and transferring the message data to application memory is wholly unrelated to a one-shot completion process of an RDMA operation.

Regarding the first argument in the Examiner's Answer, the Examiner's Answer states the following:

> Please refer to the examiner's response for Ground IV rejection of claim 17 above, for the argument presented on page 38.[53]

However, the Appellant notes that the Appellant's argument on Page 38 of the Appellant's Brief on Appeal states the following:

> The Appellant submits that the combination of Osborne, Pandya and Roach does not disclose or suggest at least the limitations of "completing an RDMA write operation using a **one-shot** <u>**completion**</u> **process** <u>**between a NIC and a driver of a host**</u>, wherein **the one-shot completion process comprises communicating a** <u>**single completion**</u>

[53] Examiner's Answer, Page 47, Lines 1-2.

> **message comprising: a send complete indication, buffer freeing status information, and an STag value**," "identifying the STag value **in a first field of a header of the single completion message**," and "validating the STag value **in the first field of the header** by identifying a bit flag or other specified value in a second field of the header," as set forth in Appellant's independent claim 25.

Nowhere in the Examiner's response for Ground IV is there any discussion regarding "identifying the STag value **in a first field of a header of the single completion message**," and "validating the STag value **in the first field of the header** by identifying a bit flag or other specified value in a second field of the header," as recited in Appellant's independent claim 25. Specifically, regarding "identifying the STag value **in a first field of a header of the single completion message**," and "validating the STag value **in the first field of the header** by identifying a bit flag or other specified value in a second field of the header," the Final Office Action acknowledges that Osborne and Pandya fails to teach the Appellant's limitations; however, the Final Office Action alleges that Roach's disclosure at Figures 8-9 and Column 8, Lines 32-67 and Column 9, Lines 1-18 remedy the deficiencies of Osborne and Pandya.[54]

Roach merely teaches placing predetermined bits indicating which is the last frame in a series of frames in a Fibre Channel frame header (FC-2 header).[55] Nowhere in Roach is there any disclosure of STag values. Further, the Final Office Action-cited Figures 8-9 of Roach merely describe a Buffer Point List Format (Figure 9) and the Buffer Pointer List Entry Format (Figure 8) for the Buffer Point List that "must exist in

[54] Final Office Action, Page 25, Line 13 – Page 26, Line 5.

[55] *See e.g.*, Roach, Abstract, Column 6, Lines 21-25 and Column 8, Lines 54-55.

contiguous physical memory."[56] In other words, Roach is wholly unrelated to identifying an STag value in a first field of a header and validating the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header as alleged in the Final Office Action, and instead is related to buffer lists stored in physical memory.

Regarding the second argument in the Examiner's Answer, the Examiner's Answer states the following:

> Regarding the appellants' argument on page 39 of the appeal brief, Figs. 7 and 33 of Pandya reference clearly show that a single completion message is also conveyed between a NIC and a driver of a host in steps 3310-3312 of Fig. 33.[57]

However, the Appellant first notes that the write operation disclosed in Figure 33 is an SCSI write, not an RDMA write. Further, 3310-3312 merely illustrates a status and sense message sent from a target to an initiator. Nowhere in Figure 33 and the supporting disclosure is there any teaching regarding a message being sent between a driver and NIC of a host, let alone "a one-shot completion process between a NIC and a driver of a host, wherein the one-shot completion process comprises communicating a single completion message comprising: a send complete indication, buffer freeing status information, and an STag value." Further, although Figure 7 at least illustrates a Remote Direct Memory Access, nowhere in Figure 7 and the supporting disclosure is

[56] Roach, Column 8, Lines 31-35.
[57] Examiner's Answer, Page 47, Lines 3-5.

there any teachings related to "a one-shot completion process between a NIC and a driver of a host, wherein the one-shot completion process comprises communicating a single completion message comprising: a send complete indication, buffer freeing status information, and an STag value."

Also, Pandya merely teaches sending a command completion and sense status once all the data has been transferred. As is well known in the art, sense status is merely error information, which is different than buffer freeing status information (i.e., information indicating that the buffers have been freed) and an STag value. As such, Pandya's mere disclosure of sending a completion message including sense status fails to teach "completing an RDMA write operation using a **one-shot <u>completion</u> process <u>between a NIC and a driver of a host</u>**, wherein **the one-shot completion process comprises communicating a <u>single completion message</u> comprising: a send complete indication, <u>buffer freeing status information</u>, <u>and</u> <u>an STag value</u>**," as recited in Appellant's independent claim 25.

Regarding the third argument in the Examiner's Answer, the Examiner's Answer states the following:

> In response to appellants' allegation on page 41 of the appeal brief, the examiner has used Roach's teachings and not a conclusory statement to reject the claim element. Furthermore, there are two different communications between a NIC and a driver of a host: one at the receiving end and the other at the sending end. It is during these communications that the pinned down buffers are to be released at each end after the message data has been successfully transferred (Please see

> Fig. 7 in Pandya and the examiner's response to rejection of claim 20 in Ground V). Just as Osborne's "Send" command includes "source address" as STag (pointer) to message buffers on the source side, likewise, Pandya's "Send completion status and sense" from the receiving host to receiving NIC will include pointer to received message buffers.[58]

However, regarding the Examiner's comment that he is using the teachings of Roach and not a conclusory statement to reject the claim element, the Appellant respectfully disagrees. Specifically, the Final Office Action alleges that "since releasing the buffers implies knowing where in the memory the buffers are, it is implied that a pointer (STag) to the message buffers is also sent with the send complete indication." In other words, although Roach does not actually teach that the send completion indication includes an STag (in fact, Roach fails to teach STags), the Examiner makes the conclusory statement that such inclusion of an STag value (when STag values are not described in Roach) is implied.

As noted in the Appellant's Brief on Appeal, MPEP 2112-IV ("Examiner must provide rationale or evidence tending to show inherency") states the following:

> **The fact that a certain result or characteristic <u>may</u> occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic.** *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) (reversed rejection because inherency was based on what would result due to optimization of conditions, not what was necessarily present in the prior art); *In re Oelrich*, 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981). "To establish inherency, the extrinsic evidence 'must make clear that **the missing descriptive matter is necessarily present in the thing described in the reference**, and that it would be so recognized by persons of ordinary skill. **Inherency, however, may not be established by probabilities or**

[58] Examiner's Answer, Page 47, Lines 6-15.

> **possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' "** *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted) (The claims were drawn to a disposable diaper having three fastening elements. The reference disclosed two fastening elements that could perform the same function as the three fastening elements in the claims. The court construed the claims to require three separate elements and held that the reference did not disclose a separate third fastening element, either expressly or inherently.).[59]

The Final Office Action fails to provide rationale or evidence showing that an STag **will necessarily need to be present** in a completion message in order to release buffers. Instead, one of ordinary skill in the art would readily understand that a sender, upon receiving an indication that the transfer is complete, would not necessarily need the receiver to identify which message buffers at the sender need to be released because such information about the sender message buffers would already be known by the sender. Further, even if a receiver was to provide a sender with information identifying the message buffers, such identification need not be using an STag value. Therefore, the rejection based on the Final Office Action's conclusory statement that "since releasing the buffers implies knowing where in the memory the buffers are, it is implied that a pointer (STag) to the message buffers is also sent with the send complete indication" cannot be maintained without evidence or rationale showing that an STag **will necessarily need to be present** in a completion message in order to release buffers. As such, the Appellant submits that Pandya fails to remedy the deficiencies of Osborne and Roach in that the combination of references clearly fails to teach a single completion message comprising an STag value.

[59] Manual of Patent Examining Procedure (MPEP) at § 2112.

Regarding the Examiner's comments referring to "two different communications between a NIC and a driver of a host: one at the receiving end and the other at the sending end," the Appellant notes that nowhere in any of the references cited by the Final Office Action is there any disclosure regarding "a single completion message comprising: a send complete indication, buffer freeing status information, and an STag value," let alone the single completion message being communicated between a driver and a NIC of a host. Instead, regarding the sending host, Pandya merely teaches communicating a message indicating that the communication session is complete up to an SCSI layer (i.e., after the completion message has passed between a driver and NIC of the sending host), which releases the buffers used for the data transfer.[60] Further, Pandya does not provide any disclosure regarding releasing buffers at the receiving host based on its "send completion status and sense."

Regarding the Examiner's comments referring to Pandya's Figure 7 and the Examiner's arguments related to claim 20 in Ground V, Pandya's disclosure at Figure 7 does not provide any teaching related to freeing buffers, let alone "completing an RDMA write operation using a **one-shot <u>completion</u> process <u>between a NIC and a driver of a host</u>**, wherein **the one-shot completion process comprises communicating a <u>single completion message</u> comprising: a send complete indication, <u>buffer freeing status information</u>, <u>and</u> <u>an STag value</u>**," as recited in Appellant's independent claim 25.

[60] Pandya, Column 13, Lines 35-48.

Regarding the fourth argument in the Examiner's Answer, the Examiner's Answer states the following:

> In response to appellants' allegation on page 44 of the appeal brief, the examiner would like to refer to Figs. 8-9 of Roach et al. reference that shows use of pointers (Buffer Starting Word Address [BSWA]) for the allocated buffers, which is what the appellants' define as STag. Furthermore, Roach et al reference is not wholly unrelated to identifying fields in FC2 (Fiber Channel 2) header, as can be seen in Fig. 7 that includes current buffer list address and different control parameters in the FC2 header.[61]

However, as discussed previously, Roach's BSWA is not an STag. Even if Roach's BSWA could be considered an STag (which it clearly is not), the BSWA is not in the FC-2 header.[62] Instead, Roach teaches that the buffer list modifier (BLM) bits are used to build an outgoing Fibre Channel frame header (i.e., FC-2 header).[63] As such, BSWA is not an STag and BSWA is not present in a first field of a header. As such, Roach's disclosure of BSWA is wholly unrelated to identifying an STag value in a first field of a header and validating the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header as alleged in the Final Office Action, and instead is related to buffer lists stored in physical memory.

Clearly, Osborne merely teaches communicating a message from a sender 50 to a receiver 52 without disclosing any completion process of an RDMA operation performed between the driver and the NIC of the host, let alone a one-shot completion

[61] Examiner's Answer, Page 47, Lines 16-21.
[62] See e.g., Roach, Figure 7, Bits 7-12.
[63] *See e.g.*, Roach, Abstract, Column 6, Lines 21-25 and Column 8, Lines 54-55.

process;[64] Pandya merely teaches communicating a message indicating that the communication session is complete up to an SCSI layer, which releases the buffers used for the data transfer;[65] and, Roach merely teaches placing predetermined bits indicating which is the last frame in a series of frames in a Fibre Channel frame header (FC-2 header).[66]

Therefore, the Appellant maintains that at least the limitations of "completing an RDMA write operation using a **one-shot completion process between a NIC and a driver of a host**, wherein **the one-shot completion process comprises communicating a single completion message comprising: a send complete indication, buffer freeing status information, and an STag value**," "identifying the STag value **in a first field of a header of the single completion message**," and "validating the STag value **in the first field of the header** by identifying a bit flag or other specified value in a second field of the header," as set forth in Appellant's independent claim 25, are not unpatentable over Osborne in view of Pandya and further in view of Roach.

Accordingly, independent claim 25 is not unpatentable over Osborne in view of Pandya and further in view of Roach and is allowable. Furthermore, the Appellant reserves the right to argue additional reasons beyond those set forth herein to support the allowability of claim 25.

[64] *See e.g.*, Osborne, Abstract; Column 4, Lines 62-64; Column 5, Lines 14 and 24-30; Column 8, Lines 16-19, 27-33 and 63-66; Column 9, Lines 1-4, 10-11 and 33-43; and Column 10, Lines 6-7.
[65] Pandya, Column 13, Lines 35-48.
[66] *See e.g.*, Roach, Abstract, Column 6, Lines 21-25 and Column 8, Lines 54-55.

CONCLUSION

For at least the foregoing reasons, the Appellant submits that claims 1-25 are in condition for allowance. Reversal of the Examiner's rejection and issuance of a patent on the application are therefore requested.

The Commissioner is hereby authorized to charge additional fee(s) or credit overpayment(s) to the deposit account of McAndrews, Held & Malloy, Account № 13-0017.

Respectfully submitted,

Date: 4-FEB-2011

By: /Philip Henry Sheridan/
Philip Henry Sheridan
Reg. No. 59,918
Attorney for Appellant

McANDREWS, HELD & MALLOY, LTD.
500 West Madison Street, 34th Floor
Chicago, Illinois 60661
(T) 312 775 8000
(F) 312 775 8100

(PHS)